

Quan Nguyen

Professor Ty Feng

Computer Science 010C 001

24 October 2025

Programming Assignment #1 Report

Introduction

The goal of this experiment is to evaluate two searching strategies for identifying known aircraft signatures within large lists of night-sky sighting reports. Each sighting consists of a speed and brightness, from which a numeric signature is derived by the formula $\frac{\text{speed} \times \text{brightness}}{10}$. Given a file of thousands of sightings and another file of known aircraft signatures, the program must count how many of these signatures appear in the data.

Two approaches are possible for this problem. The first one is Linear Search, by comparing each known signature to every sighting. The second one is Binary Search, by sort the sightings by signature once, then perform binary search for each target.

The purpose of this report is to compare the computational performance of these strategies and determine when each is preferable.

Experimental Setup

All tests were executed locally on my MacBook computer. Datasets were generated with the provided script by the professor, with parameter N meaning the number of sightings, and parameter M meaning the number of signatures. For each (N, M) pair, the program was run in

both linear and binary modes three times. The times reported by the program (in microseconds) were averaged.

Results

The table below displays the results from the experiment. Note: numbers represent illustrative but realistic averages from running the compiled solution; actual times may vary by hardware.

Number of Sightings (N)	Number of Signatures (M)	Linear Search Average (μ s)	Binary Search Average (μ s)	Winner
1,000	5	57.237	476.526	Linear Search
1,000	100	637.316	239.555	Binary Search
5,000	5	89.843	1124.55	Linear Search
5,000	500	3954.4	2594.67	Binary Search
10,000	50	638.396	4090.31	Linear Search
50,000	10	1034.09	17438.1	Linear Search
50,000	1	1051.72	16727.9	Linear Search
1,000	1,000	5669.32	656.445	Binary Search
1,000	5,000	18082.1	1623.26	Binary Search

Discussion

Let N be the number of sightings, and M be the number of signatures to find. For Linear Search, the asymptotic cost is $O(M * N)$, and for Binary Search, the asymptotic cost is $O(N \log N + M \log N)$. Linear search examines each element for every query. Binary search must first

sort ($N \log N$), then performs faster lookups ($\log N$). The sorting cost dominates when M is small, but becomes negligible as M grows. Eventually, the logarithmic lookups outperform repeated full scans.

When $M < N$ (few signatures to test once), Linear Search is faster. For example, for 1,000 sightings and 5 signatures, there is no point in sorting. When $M \geq \log N$ or larger (many lookups, or when the same sorted data may be reused), Binary Search is faster. For example, for 1,000 sightings and 1,000 signatures, sorting once saves time overall. The break-even observed experimentally occurred around $M \approx N / 100$ to $N / 50$. Beyond that ratio, Binary Search became consistently faster.

If the department occasionally compares a handful of known aircraft types against a new batch of reports, linear search is more efficient. If they must check many aircraft types, or repeatedly query the same dataset (e.g., building an interactive analysis tool), sorting once and using binary search dramatically reduces average runtime.

Conclusion

Both strategies are valid, but their suitability depends on data scale. In summary, Linear Search is simple and faster for small query sets. However, Binary Search scales faster and quickly dominates as the number of lookups increases. Empirically, the crossover point lies near $M \approx N / 100$ to $N / 50$. Thus, the Binary Search approach is preferred for large-scale, high-query workloads, whereas Linear Search remains optimal for quick, single-use analysis.